AD-A103 098 KANSAS STATE UNIV MANHATTAN DEPT OF COMPUTER SCIENCE F/G 9/2
RESEARCH IN FUNCTIONALLY DISTRIBUTED COMPUTER SYSTEMS DEVELOPME--ETC(U)
APR 77 F J MARYANSKI, V E WALLENTINE DAAG29-76-G-0108
UNCLASSIFIED CS-77-7 NL

1 OF 1
A
103098

END
DATE
FILMED
9-81
DTIC

AIRMICS
Army Institute for Research in Management Information and Computer Science
313 Calculator Bldg.
GA Institute of Technology
Atlanta, GA 30332

Technical Report

RESEARCH IN FUNCTIONALLY DISTRIBUTED COMPUTER SYSTEMS DEVELOPMENT

Kansas State University

Virgil Wallentine
Principal Investigator

VOLUME XV

PERFORMANCE OF MULTI-PROCESSOR BACKEND DATA BASE SYSTEMS

U.S. ARMY COMPUTER SYSTEMS COMMAND FT BELVOIR, VA 22060

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

9 Interim rept.

REPORT DOCUMENTATION PAGE

READ INSTRUCTIONS BEFORE COMPLETING FORM

1. ~~REPORT NUMBER~~

2. GOVT ACCESSION NO. AD-A103 098

3. RECIPIENT'S CATALOG NUMBER

4. 6 Research in Functionally Distributed Computer Systems Development. Volume XV. PERFORMANCE OF MULTI-PROCESSOR BACKEND DATA BASE SYSTEMS.

5. TYPE OF ~~REPORT & PERIOD COVERED~~ Iterim

6. ~~PERFORMING~~ ORG. REPORT NUMBER 14 CS-77-7

7. ~~AUTHOR(s)~~ 10 J. Fred Maryanski, Virgil Wallentine

8. CONTRACT OR GRANT NUMBER(s) 15 DAAG 29-76-G-Ø1Ø8

9. PERFORMING ~~ORGANIZATION~~ NAME AND ADDRESS
Kansas State University
Department of Computer Science
Manhattan, KS 66506

10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS

11. CONTROLLING OFFICE NAME AND ADDRESS
US Army Research Office
P O Box 12211
Research Triangle Park, NC 27700

12. ~~REPORT DATE~~ 11 Apr[redacted]77

13. NUMBER OF PAGES 17 pages

14. MONITORING AGENCY NAME & ADDRESS(if different from Controlling Office)
US Army Computer Systems Command
Attn: CSCS-AT
Ft. Belvoir, VA 22060 12 20

15. SECURITY CLASS. (of this report) Unclassified

15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

Approved for public release; distribution unlimited.

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

The findings in this report are not to be construed as an official Department of the Army position, unless so designated by other authorized documents.

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

DBMS
Back-end Computer

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

-over-

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE

Unclassified 391123

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

-ABSTRACT-

The results of a simulation study intended to determine the circumstances under which it is beneficial to operate a data base management system with a multi-processor backend are presented. The basic concept of backend data base management systems and multi-processor backend systems are provided as background material. The general structure of the simulation model which has been implemented in GPSS is outlined. The results of the study indicate that the amount of CPU activity required by the data base management system is a determining factor with respect to the need for a multi-processor backend.

PERFORMANCE OF MULTI-PROCESSOR

BACK-END DATA BASE SYSTEMS[1]

TR CS 77-07

April 1977

Fred J. Maryanski
Computer Science Department
Kansas State University
Manhattan, Kansas
66506

[1]The work reported herein was partially supported by the United States Army Computer Systems Command Grant No. DAAG-29-76-G-108.

Published in Proceedings Conference on Information Sciences and Systems, The John Hopkins University, Baltimore, Md. Apr. 1977.

Abstract

The results of a simulation study intended to determine the circumstances under which it is beneficial to operate a data base management system with a multi-processor back-end are presented. The basic concepts of back-end data base management systems and multi-processor back-end systems are provided as background material. The general structure of the simulation model which has been implemented in GPSS is outlined. The results of the study indicate that the amount of CPU activity required by the data base management system is a determining factor with respect to the need for a multi-processor back-end.

1. Introduction

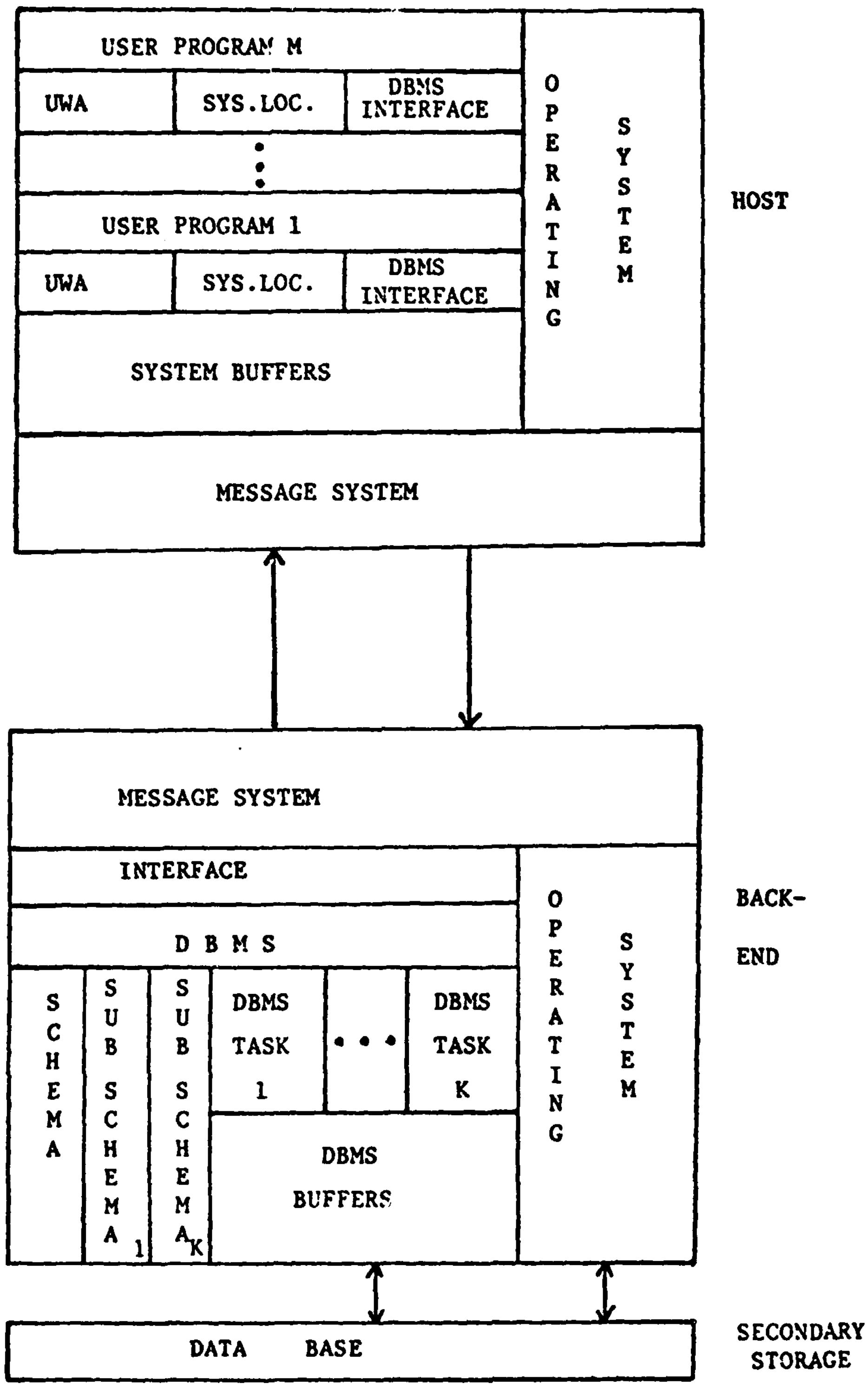
As the utilization of data base management systems (DBMS) increases, a proportionate rise in the demand for system resources is observed. Often this demand exceeds the capacity of the central computer of a data processing installation. One practical and economic method of increasing system capacity without an expensive upgrade is to attach a minicomputer to the central computer to share the data base load. The resultant configuration is known as a back-end DBMS.

Typically, a back-end DBMS consists of one processor to execute application programs and one to perform the data base functions. It is possible to distribute the data base functions over several processors in a multi-processor back-end DBMS. A simulation study has been conducted to determine the circumstances under which a multi-processor back-end DBMS provides performance superior to a single processor back-end DBMS.

In order to provide the proper background for the interpretation of the simulation results, the essential concepts and basic structure of back-end data base systems are presented.

2. Back-End Data Base Management Systems

The principal idea of a back-end data base management system is that all data base operations are performed by a computer that is distinct from the processor executing the application programs requesting the operations. The concept of a back-end DBMS was initially proposed by Canaday, et al [1] and has been studied by several other researchers [2-10]. The processor on which the application programs reside is known as the host processor. The computer controlling the access to the data base is a back-end machine. Figure 1 illustrates the distribution of software in a back-end DBMS.



Back-End DBMS Software Distribution

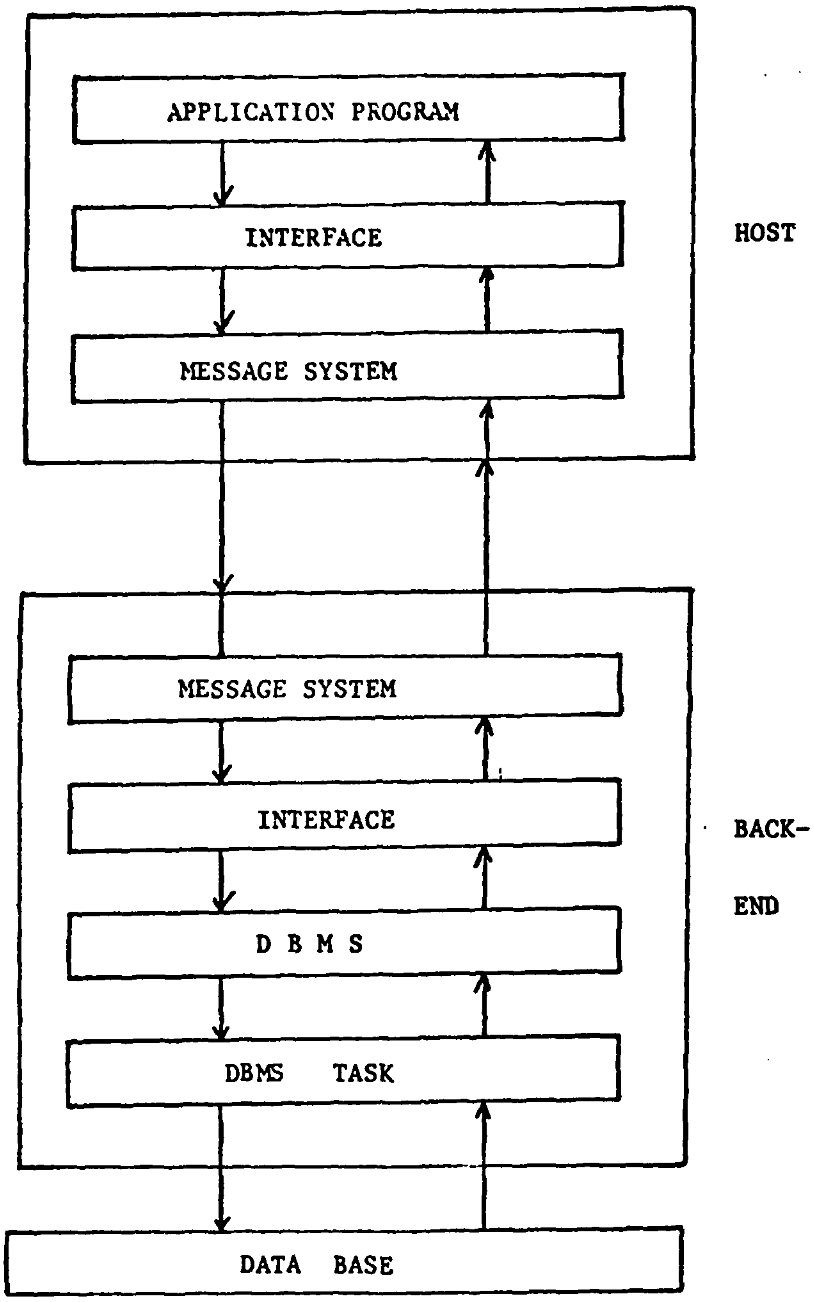
Figure 1

As shown in Figure 1, the basic philosophy of a back-end DBMS is to minimize the resources of the host processor required by the data base system. Other than the application programs, the only data base system requirements are an interface routine, buffer space, and a message system. The message system is a communications package that controls the exchange of information between the host and back-end machines. The DBMS interface routine is called whenever a data base operation is requested by an application program. The interface formats the request into a message and then calls the message system to transmit the request to the data base system software on the back-end computer.

The back-end machine contains all of the functional DBMS software. The actual execution of the data base operation is carried out by one of several DBMS tasks. It has been demonstrated [11] that in order for a back-end DBMS to provide any performance benefits, both the host and back-end processors must be multi-programmed. The DBMS control software on the back-end initiates and manages the DBMS tasks. The back-end processor also holds the logical description of the entire data base, the schema, and the logical description of the portion of the data base available to a particular application program, the sub-schema. All data base operations are validated against the sub-schema and schema by the DBMS control software before being passed to a DBMS task for execution. Figure 2 portrays the communication sequence occurring when an application program issues a data base request.

The feasibility of a back-end DBMS in a data processing environment has been demonstrated in a prior study [3]. The results of that study indicate that a back-end DBMS provides the following advantages over a single machine system:

1. The host memory allocated to the DBMS is significantly reduced.



Communication Flow in Back-End DBMS

Figure 2

2. The amount of memory per application required in the host machine is reduced.

3. Operating system overhead is decreased on the host system since many operating system functions are assumed by the back-end machine.

4. By transferring tasks to the back-end minicomputer, the advantages of minicomputers are realized--i.e., they are less expensive per instruction to operate than large mainframes.

5. Integrity is improved, since the computers can oversee one another.

6. Data base protection is enhanced, since control of the data base is independent of the host computer and its memory space.

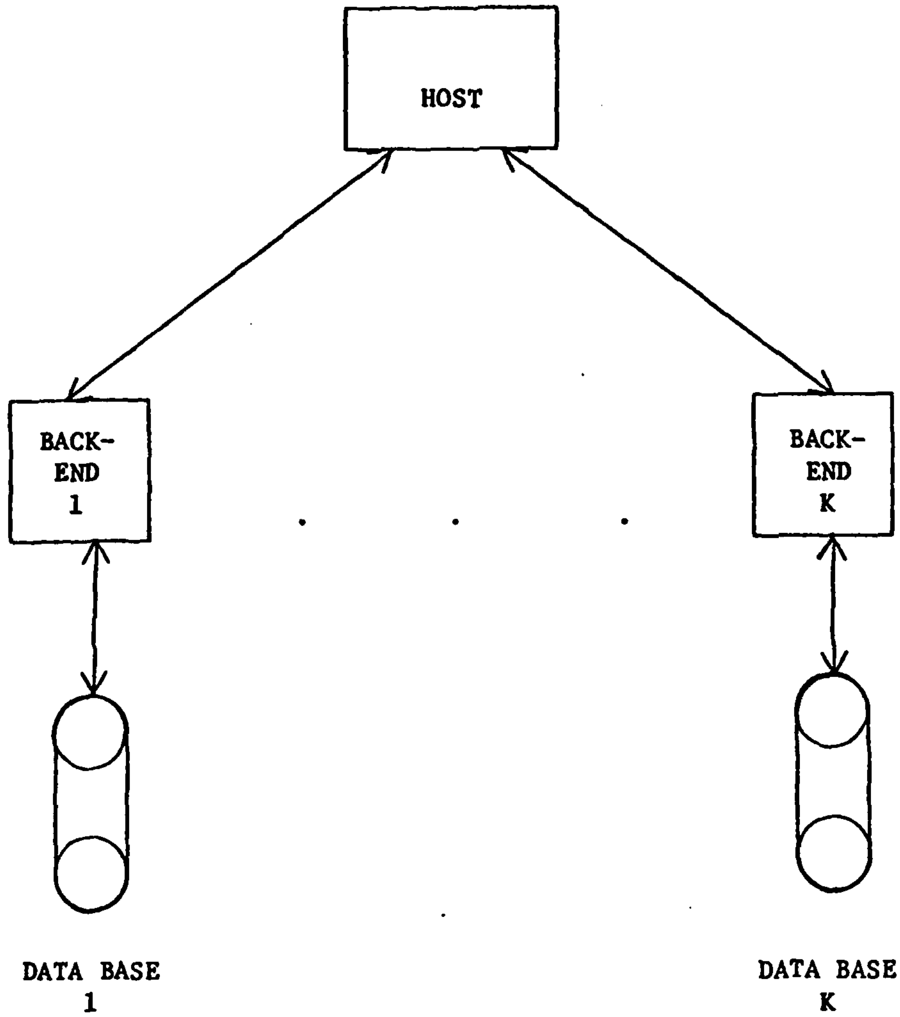
7. The system has a modular organization which provides direct and indirect benefits from the standpoints of maintenance and software development.

This study extends the notion of a back-end DBMS to allow several machines to perform the back-end function. This configuration has been proposed by Lowenthal [2] and is known as a multi-processor back-end DBMS. Figure 3 depicts the general structure of a multi-processor back-end DBMS.

The software structure of a back-end processor in a multi-processor back-end DBMS is identical to the structure in a single back-end configuration. The host processor must contain sufficient software to determine the back-end processor to execute a particular data base operation. As indicated in Reference [12], the back-end processor can be determined from the identifiers of the data being accessed.

3. The Simulation Model

The model used to describe a multi-processor back-end DBMS is an extension of the model developed by Maryanski and Wallentine [11] for a single processor back-end DBMS. Implementation was done using GPSS. A very high-level descrip-



Multi-Processor Back-End DBMS

Figure 3

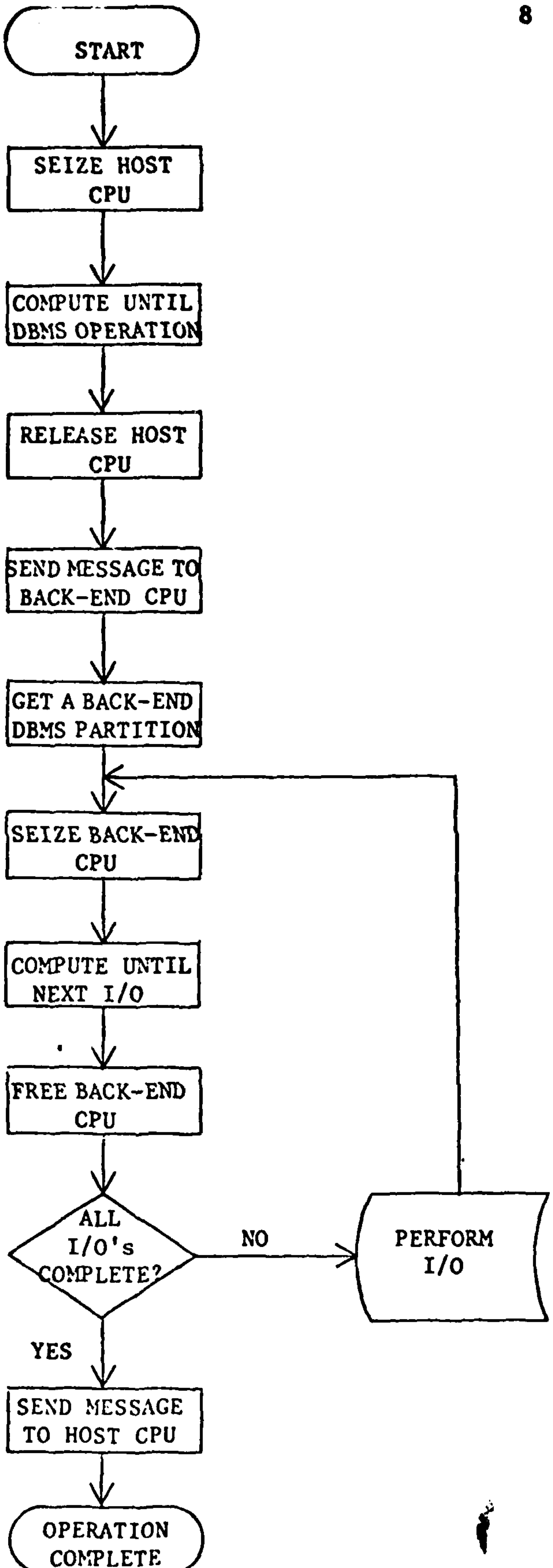
tion of the simulation of a data base operation is presented in Figure 4.

The model can be parameterized to describe any distributed data base management system with arbitrary numbers of host and back-end processors. In order to describe a configuration, the user must provide information on the intermachine linkages, relative processing capabilities, number of disks associated with each processor, and the number of data base partitions to be supported by each machine. For the simulation experiment described here, the hardware configuration portrayed in Figure 5 was modeled. This configuration is composed of computers in the laboratory of the Computer Science Department at Kansas State University.

The job mix can be characterized by the parameter indicating the amount of computation by the host CPU between data base requests. For this experiment, the average values were set at 1, 13, and 130 milliseconds to indicate light, moderate, and heavy computational activity (or conversely, heavy, moderate, and light data base activity). Parameters indicating the number of I/O operations per data base request and the amount of computation by the back-end processor between I/O operations can be used to further characterize the data base environment. For the experiment reported here, these parameters were not varied. The values of the parameters were set to describe an environment in which data base requests required very little computation in the back-end and the majority (56%) of which did not require a physical I/O. A data base system with batch updating and simple inquiries fits this characterization.

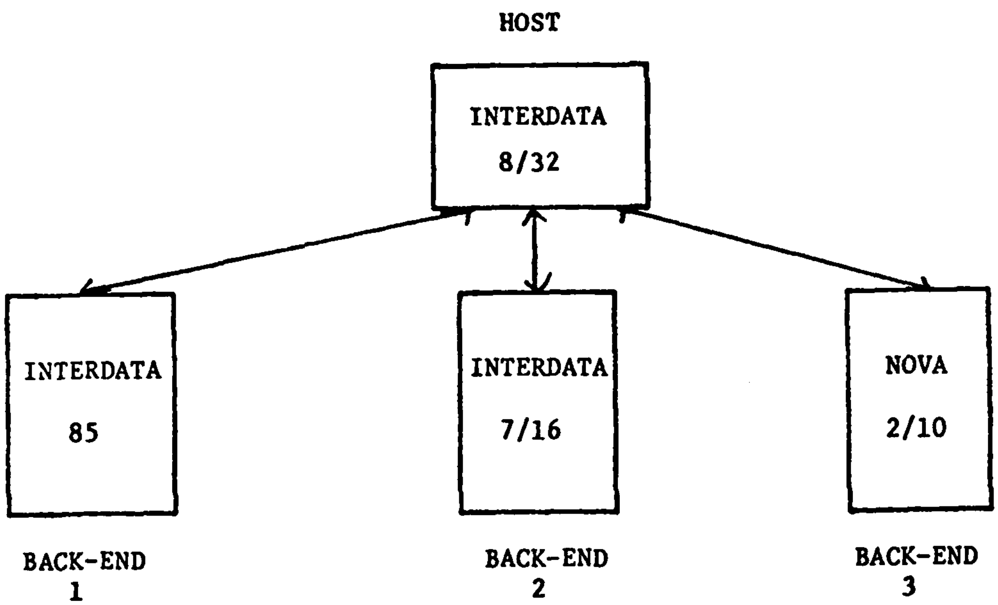
4. Results

The object of the simulation was to determine, for the particular environment, the performance effects of adding additional back-end processors to a back-end DBMS. Performance is measured in terms of the number of data base



Simulation of Data Base Operation

Figure 4



Hardware Configuration

Figure 5

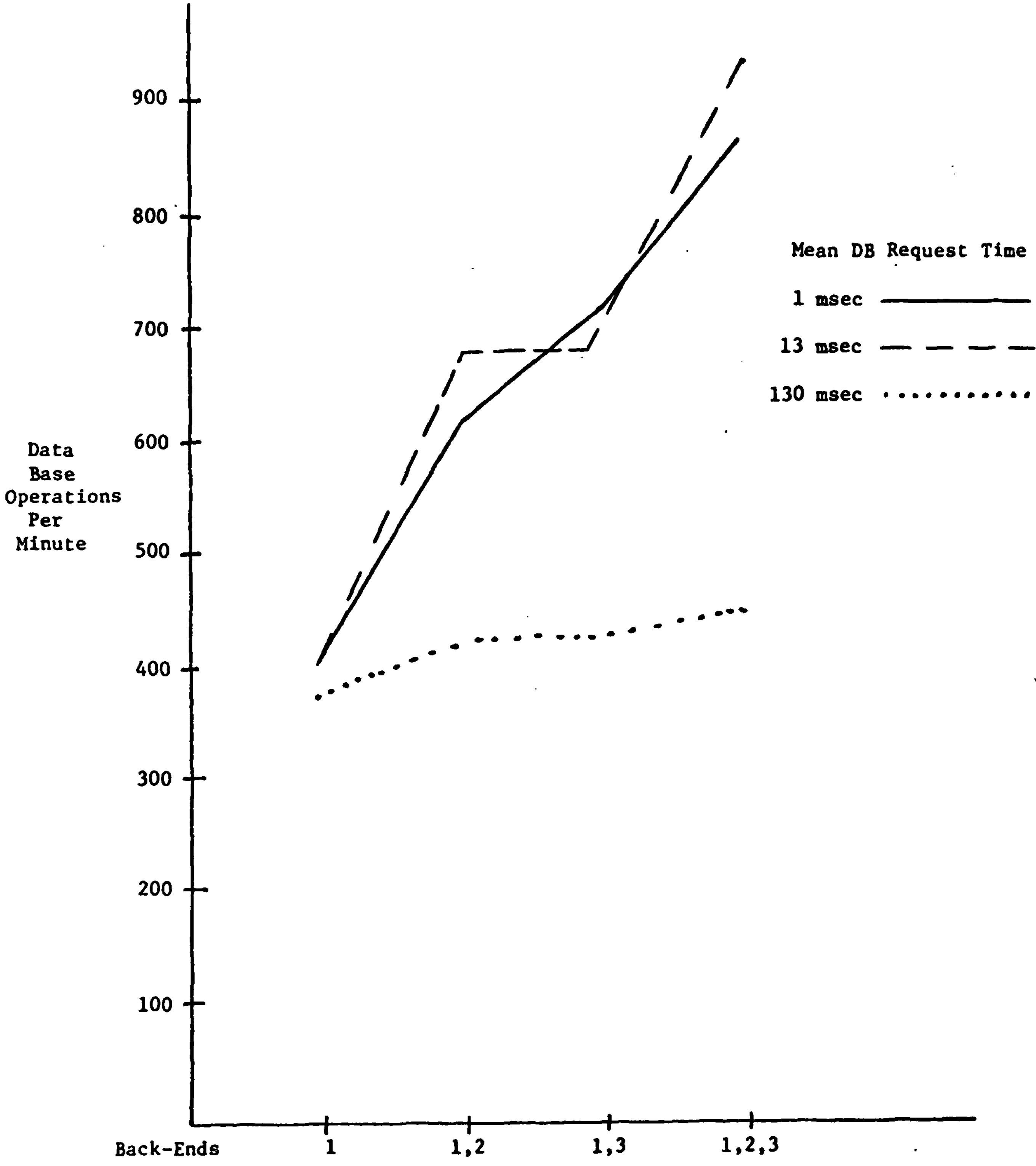
operations completed per minute. Of the back-end processors shown in Figure 5, Back-Ends 1 and 3 are assumed to be of approximately equal computational power, while Back-End 2 is assumed to be significantly less powerful. Computational power was roughly measured by relative instruction speed.

The most important parameter with respect to performance is the amount of computation between host data base requests. In all situations, with various numbers of host and back-end partitions, when the mean inter-data base requests time was set at 1 or 13 milliseconds, a significant performance increase was noted with any of the multi-processor back-end configurations. However, with a mean inter-data base request time of 130 milliseconds, the performance benefits of the multi-processor back-end configurations were insubstantial. Figures 6-8 provide a sampling of the simulation results.

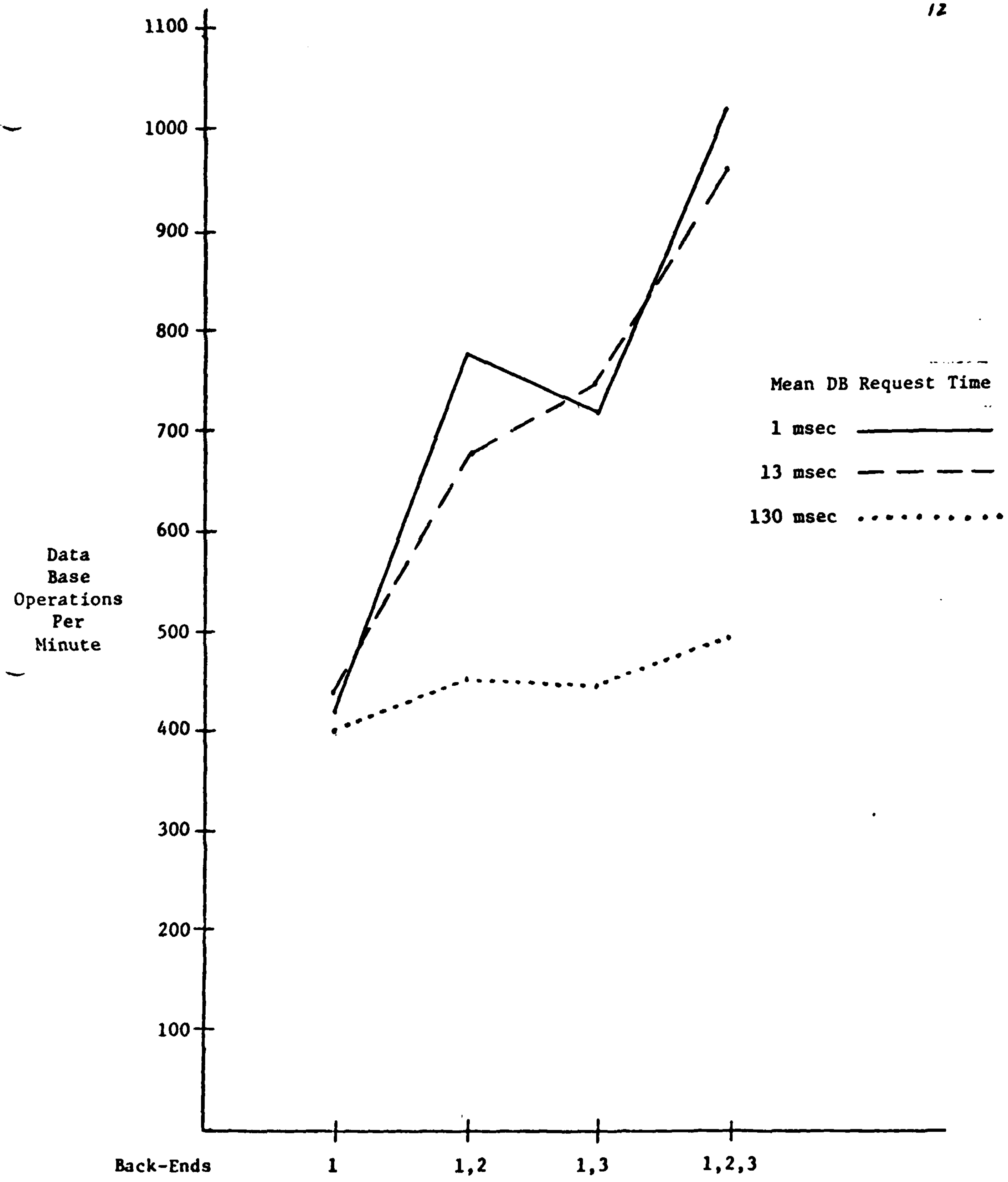
5. Analysis

The simulation results indicate that for moderate to heavy data base I/O activity, a multi-processor back-end DBMS has the potential for increasing performance in a batch update, simple inquiry environment. A "typical" commercial data processing facility has an environment similar to that used in the model. The feasibility of a single-processor back-end DBMS has been evaluated favorably in previous work [3]. The results presented here indicate that if increased performance is desired from a single processor back-end DBMS in a data processing installation, the addition of another back-end processor is a reasonable strategy if the data base demand is moderate to heavy.

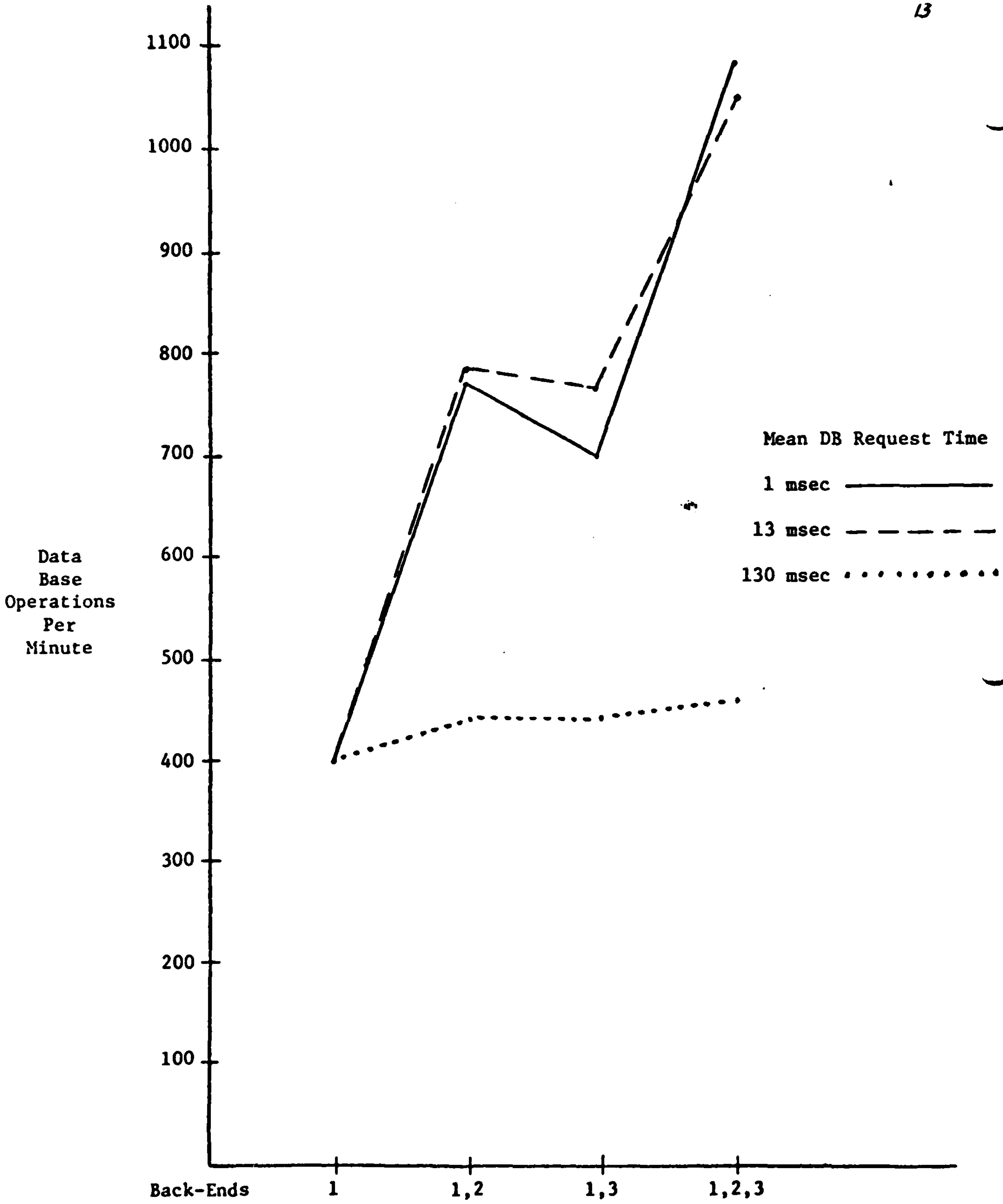
Observation of Figures 6-8 indicates no clear performance superiority between configurations 1, 2, and 1, 3. Other results not shown in the figures indicate that in a single processor back-end DBMS, Back-End 1 and



PERFORMANCE EFFECTS OF MULTI-PROCESSOR BACK-END
10 Host Partitions, 2 Partitions Per Back-End
Figure 6



PERFORMANCE EFFECTS OF MULTI-PROCESSOR BACK-END
30 Host Partitions, 2 Partitions Per Back-End
Figure 7



PERFORMANCE EFFECTS OF MULTI-PROCESSOR BACK-END
30 Host Partitions, 9 Partitions Per Back-End
Figure 8

Back-End 3 give approximately equal performance. These findings lead to the conclusion that in the environment simulated, computational speed of the back-end machine is not an important performance factor. However, preliminary results in a follow-on simulation experiment indicate that if the computation time in the back-end per data base operation is increased, the computational speed of the back-end machine can become a determinant of system performance. In terms of the data base environment, an increase in back-end computation per data base operation corresponds to an increase in complexity of the data base operations.

6. Conclusion

The simulation results indicate that for many data processing installations the progressive addition of back-end data base machines is a viable long-term plan for efficient and economical data base management. Further investigations are required to determine necessary capabilities of back-end data base computers in varying data base environments.

7. References

1. Canaday, R.E., et al., A Back-End Computer for Data Base Management, Comm. ACM, Vol. 17, No. 10, Oct., 1974, pp. 575-582.
2. Lowenthal, E.I., The Back-End Computer, MRI Systems Corp. P.O. Box 9968, Austin, Tex. 78766, Apr., 1976.
3. Maryanski, F.J., Fisher, P.S. and Wallentine, V.E., Evaluation of Conversion to a Back-End Data Base Management System, Proc. ACM Annual Conf., Oct., 1976, pp. 293-297.
4. Everest, G.C., The Futures of Data Base Management, Proc. ACM SIGMOD Workshop, May, 1974, pp. 445-462.
5. Fry, J.P. and Deppe, M.E., Distributed Data Bases: A Summary of Research, Computer Networks, Vol. 1, No. 2, 1976.

6. Fry, J.P. and Sibley, E.H., Evolution of Data Base Management Systems, Computing Surveys, Vol. 8, No. 1, Mar., 1976, pp.7-42.

7. Marill, T. and Stern, D., The Datacompter -- A Network Data Utility, Proc. National Computer Conf., Vol. 44, May, 1975, pp. 389-395.

8. Ozkarahan, E.A., Schuster, S.A. and Smith, K.E., RAP -- An Associative Processor for Data Base Management, Proc. National Computer Conf., Vol. 44, May, 1975, pp. 379-387.

9. Witney, K.M., Fourth Generation Data Management Systems, Proc. National Computer Conf., Vol. 42, 1973, pp. 239-244.

10. Baum, R.I. and Hsiao, D.K., Data Base Computers -- A Step Towards Data Utilities, IEEE Trans. on Comp., Vol. C-25, No. 12, Dec., 1976, pp. 1254-1259.

11. Maryanski, F.J. and Wallentine, V.E., A Simulation Model of a Back-End Data Base Management System, Proc. Pittsburgh Modeling and Simulation Conf., Apr., 1976.

12. Maryanski, F.J., Fisher, P.S. and Wallentine, V.E., A User-Transparent Mechanism for the Distribution of a CODASYL Data Base Management System, TR 76-22, Computer Science Dept., Kansas State University, Manhattan, Ks. 66506, Dec., 1976.